



Scope e inicialización

Transcripción

[00:00] Ya somos un poco más expertos en cómo podemos optimizar nuestras variables, nuestras condicionales, dejar el código mucho más legible. Y ahora vamos a ver el tema de scope. Scope es el alcance de las variables. Entonces, vamos a basarnos en este mismo código, pero para que quede el ejemplo de cómo vamos progresando, vamos a crear igual otra clase.

[00:31] Igual vamos a new class y ahora vamos a cambiar y vamos a poner EjemploScope. Finish. ¿Qué es el scope? El scope básicamente es todo lo que está delimitado por llaves. Viniendo de aquí, vamos a copiar nuestro método Main, vamos a hacerle un "Ctrl + C", "Ctrl + V" y en este caso el scope del método Main va desde aquí, desde la línea 4, hasta la línea 20 porque él aquí cierra las llaves.

[01:13] Lo mismo pasa digamos con las variables. Ya tenemos claro por ejemplo que si yo quisiera imprimir aquí es pareja, antes de declarar la variable no va a compilar porque me va a decir "yo no conozco quién es es pareja". Entonces, hasta ahí todo bien. El compilador ya nos está alertando. Vamos a maximizar esto. Perfecto.

[01:41] Y a modo de prueba vamos también a comentar esto de aquí. Comentamos ahí y vamos a evaluar si cantidad de personas es mayor que 1, entonces es pareja va a ser igual a true. Si no, else es pareja es igual a false. Perfecto. Damos un enter aquí, y nuevamente vemos que es pareja no está compilando aquí en estas dos líneas y aquí tampoco. ¿Por qué? Porque no hemos declarado la variable es pareja.

[02:34] Él no conoce esta variable de aquí. Sería prácticamente el mismo caso que ocurrió aquí. Ahora si yo declaro aquí boolean es pareja, él compila, pero aún así, en las siguientes líneas no compilan. ¿Por qué no sucede esto? Porque la variable es pareja está declarada dentro del alcance de este if. ¿Hasta dónde llega el alcance de este if? Llega desde que abrimos sus llaves hasta que las cerramos.

[03:09] Y en este caso, es pareja está fuera del alcance de este if. ¿Aquí nosotros podríamos hacer esta declaración, si la hiciéramos aquí a este nivel? Puede entrar. Va a compilar perfecto. ¿Por qué? Porque es pareja existe en este scope, en este alcance. Es pareja está dentro de este contexto de if, por lo tanto va a compilar tranquilamente. Fuera de este contexto no va a compilar.

[03:40] Ahora yo podría también declarar aquí: boolean es pareja. Compilo aquí, y de igual forma no va a compilar en esto porque están fuera del scope de if y de else. Están totalmente fuera de estos dos alcances. Entonces yo podría hacer algo como esto. Voy a declarar solamente un boolean que es pareja. Si es pareja, entonces pasa a cumplir una condición o no.

[04:11] Como ya está declarado es pareja en un contexto mayor al contexto que está dentro de if y else, va a dar un error de compilación porque esta variable ya está siendo usada. Por ejemplo aquí el error dice: Duplicate local variable es pareja. Entonces, vamos a borrar esto para usar la misma variable. Y listo, ahora vemos qué compila.

[04:36] Decimos es pareja, sí. Cantidad de personas mayor a 1 true, si no, me va a dar false. Pero recordemos que todo boolean por defecto es false. Y yo podría hacer esto. Si es pareja, es true. Si no, él debería quedar como false. Ahora, nuevamente aquí no compila. No está compilando en estas dos líneas. Analicemos el código.

[05:04] Declaré mi variable, está en el mismo contexto que estas condiciones y estoy asignando un valor a ella, que es true. ¿Por qué no compila? La respuesta

es: porque recuerden que toda variable tiene que ser inicializada. Por ejemplo, si yo quisiera aquí hacer un sysout de es pareja, él no me va a dejar compilar aquí. ¿Por qué? Porque aquí la variable no ha sido inicializada.

[05:38] Y ya sabemos que Java no permite usar una variable que no está inicializada. Estamos inicializando es pareja aquí si y solo si la cantidad de personas es mayor a 1. Entonces el compilador dice: "si existe la mínima posibilidad de que es pareja sea false, entonces tú nunca vas a inicializar es pareja, por lo tanto yo no voy a compilar ese código". Es así como funciona en Java.

[06:08] Entonces, podríamos hacer tranquilamente la parte de else, nuevamente, si no, es pareja es igual a false. Y con eso ya satisfacemos la condición de que sea cual sea el caso, sea la condición true o false, estamos inicializando es pareja. Y al mismo tiempo, también es pareja está declarado en el contexto más grande, que es pues el contexto del método Main.

[06:39] De esta forma entonces ya tengo más claro cómo es que trabajan los contextos de las variables y en qué casos no podemos utilizar digamos una variable sabiendo pues si es que ha sido inicializada o no, o si es que existe la mi oportunidad de que esa variable no sea inicializada.